<https://www.sql-practice.com/>

SQL-Practice Website

---------------------------------------------------------------------------------------------

**SQL DATABASE**

SQL SCHEMA

TABLES:

|  |  |  |
| --- | --- | --- |
| patients | | |
| Key Type | Column Name | Data Type |
| Primary Key | patient\_id | INT |
|  | first\_name | TEXT |
|  | last\_name | TEXT |
|  | gender | CHAR(1) |
|  | birth\_date | DATE |
|  | city | TEXT |
| Foreign Key | province\_id | CHAR(2) |
|  | allergies | TEXT |
|  | height | INT |
|  | Weight | INT |

|  |  |  |
| --- | --- | --- |
| admissions | | |
| Key Type | Column Name | Data Type |
| Foreign Key | patient\_id | INT |
|  | admission\_date | DATE |
|  | discharge\_date | DATE |
|  | diagnosis | TEXT |
| Foreign Key | attending\_doctor\_id | INT |

|  |  |  |
| --- | --- | --- |
| doctors | | |
| Key Type | Column Name | Data Type |
| Foreign Key | doctor\_id | INT |
|  | first\_name | TEXT |
|  | last\_name | TEXT |
|  | specialty | TEXT |

|  |  |  |
| --- | --- | --- |
| province\_names | | |
| Key Type | Column Name | Data Type |
| Foreign Key | province\_id | CHAR(2) |
|  | province\_name | TEXT |

**Questions & Answers**

**Easy Questions**

**Questions 1:**

Show first name, last name, and gender of patients who's gender is 'M'

**Solution:**

SELECT first\_name, last\_name, gender

FROM patients

WHERE gender= 'M';

**Questions 2:**

Show first name and last name of patients who does not have allergies. (null)

**Solution:**

SELECT

first\_name,

last\_name

FROM patients

WHERE allergies IS NULL;

**Questions 3:**

Show first name of patients that start with the letter 'C'

**Solution:**

SELECT

first\_name

FROM patients

WHERE first\_name LIKE 'C%';

**or,**

SELECT first\_name

FROM patients

WHERE substring (first\_name, 1, 1) = 'C';

**Questions 4:**

Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)

**Solution:**

SELECT

first\_name,

last\_name

FROM patients

WHERE weight BETWEEN 100 AND 120;

**or,**

SELECT

first\_name,

last\_name

FROM patients

WHERE weight >= 100 AND weight <= 120;

**Questions 5:**

Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'

**Solution:**

UPDATE patients

SET allergies = 'NKA'

WHERE allergies IS NULL;

**Questions 6:**

Show first name and last name concatinated into one column to show their full name.

**Solution:**

SELECT (first\_name || ' ' || last\_name) AS full\_name

FROM patients;

**or,**

SELECT

CONCAT (first\_name, ' ', last\_name) AS full\_name

FROM patients;

**Questions 7:**

Show first name, last name, and the full province name of each patient.

Example: 'Ontario' instead of 'ON'

**Solution:**

SELECT patients.first\_name, patients.last\_name, province\_names.province\_name

FROM patients

JOIN province\_names

WHERE patients.province\_id = province\_names.province\_id

GROUP BY patient\_id;

**Questions 8:**

Show how many patients have a birth\_date with 2010 as the birth year.

**Solution:**

SELECT COUNT (patient\_id) AS total\_patients

FROM patients

WHERE birth\_date LIKE '2010%';

**or,**

SELECT count(first\_name) AS total\_patients

FROM patients

WHERE

birth\_date >= '2010-01-01'

AND birth\_date <= '2010-12-31’;

**Questions 9:**

Show the first\_name, last\_name, and height of the patient with the greatest height.

**Solution:**

SELECT

first\_name,

last\_name,

MAX (height) AS height

FROM patients;

**or,**

SELECT

first\_name,

last\_name,

height

FROM patients

ORDER BY height DESC

LIMIT 1;

**or,**

SELECT

first\_name,

last\_name,

height

FROM patients

WHERE height = (

SELECT max(height)

FROM patients

)

**Questions 10:**

Show all columns for patients who have one of the following patient\_ids:

1,45,534,879,1000

**Solution:**

SELECT \*

FROM patients

WHERE patient\_id IN (1,45,534,879,1000);

**Questions 11:**

Show the total number of admissions.

**Solution:**

SELECT COUNT (admission\_date) AS total\_addmissions

FROM admissions;

**Questions 12:**

Show all the columns from admissions where the patient was admitted and discharged on the same day.

**Solution:**

SELECT \*

FROM admissions

WHERE admission\_date = discharge\_date;

**Questions 13:**

Show the patient id and the total number of admissions for patient\_id 579.

**Solution:**

SELECT patient\_id, COUNT (admission\_date) AS total\_admissions

FROM admissions

WHERE patient\_id = 579;

**Questions 14:**

Based on the cities that our patients live in, show unique cities that are in province\_id 'NS'?

**Solution:**

SELECT DISTINCT city as unique\_cities

FROM patients

WHERE province\_id = 'NS';

**or,**

SELECT city

FROM patients

GROUP BY city

HAVING province\_id = 'NS';

**Questions 15:**

Write a query to find the first\_name, last name and birth date of patients who have height more than 160 and weight more than 70.

**Solution:**

SELECT first\_name, last\_name, birth\_date

FROM patients

WHERE height > 160 AND weight >70;

**Questions 16:**

Write a query to find list of patients first\_name, last\_name, and allergies from Hamilton where allergies are not null.

**Solution:**

SELECT first\_name, last\_name, allergies

FROM patients

WHERE city = 'Hamilton' AND allergies IS NOT NULL;

**Questions 17:**

Based on cities where our patient lives in, write a query to display the list of unique cities starting with a vowel (a, e, i, o, u). Show the result order in ascending by city.

**Solution:**

SELECT DISTINCT city

FROM patients

WHERE (city like 'a%') OR

(city like 'e%') OR

(city like 'i%') OR

(city like 'o%') OR

(city like 'u%')

ORDER BY city ASC;

**Medium Questions**

**Questions 1:**

Show unique birth years from patients and order them by ascending.

**Solution:**

SELECT YEAR (birth\_date) AS birth\_year

FROM patients

GROUP BY YEAR (birth\_date);

**Questions 2:**

Show unique first names from the patients table which only occurs once in the list.

For example, if two or more people are named 'John' in the first\_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output.

**Solution:**

SELECT first\_name

FROM patients

GROUP BY first\_name

HAVING COUNT (first\_name) = 1;

**or,**

SELECT first\_name

FROM (

SELECT

first\_name,

count(first\_name) AS occurrencies

FROM patients

GROUP BY first\_name )

WHERE occurrencies = 1

**Questions 3:**

Show patient\_id and first\_name from patients where their first\_name start and ends with 's' and is at least 6 characters long.

**Solution:**

SELECT patient\_id, first\_name

FROM patients

WHERE first\_name LIKE 's\_\_\_\_%s';

**or,**

SELECT

patient\_id,

first\_name

FROM patients

WHERE

first\_name LIKE 's%s'

AND len(first\_name) >= 6;

**or,**

SELECT

patient\_id,

first\_name

FROM patients

WHERE

first\_name like 's%'

AND first\_name like '%s'

AND len(first\_name) >= 6;

**Questions 4:**

Show patient\_id, first\_name, last\_name from patients whos diagnosis is 'Dementia'.

Primary diagnosis is stored in the admissions table.

**Solution:**

SELECT

patients.patient\_id,

patients.first\_name,

patients.last\_name

FROM patients

JOIN admissions on patients.patient\_id = admissions.patient\_id

WHERE diagnosis = 'Dementia';

**or,**

SELECT

patient\_id,

first\_name,

last\_name

FROM patients

WHERE patient\_id IN (

SELECT patient\_id

FROM admissions

WHERE diagnosis = 'Dementia'

);

**Questions 5:**

Display every patient's first\_name.

Order the list by the length of each name and then by alphabetically.

**Solution:**

SELECT first\_name

FROM patients

ORDER BY LENGTH (first\_name), first\_name ASC;

**Questions 6:**

Show the total amount of male patients and the total amount of female patients in the patients table.

Display the two results in the same row.

**Solution:**

SELECT SUM (gender=='M') AS male\_count, SUM (gender=='F') AS female\_count

FROM patients;

**or,**

SELECT

(SELECT COUNT (\*) FROM patients WHERE gender='M') AS male\_count,

(SELECT COUNT (\*) FROM patients WHERE gender='F') AS female\_count;

**Questions 7:**

Show first and last name, allergies from patients which have allergies to either 'Penicillin' or 'Morphine'. Show results ordered ascending by allergies then by first\_name then by last\_name.

**Solution:**

SELECT first\_name, last\_name, allergies

FROM patients

WHERE allergies='Penicillin' OR allergies = 'Morphine'

ORDER BY allergies, first\_name, last\_name ASC;

**or,**

SELECT

first\_name,

last\_name,

allergies

FROM patients

WHERE

allergies IN ('Penicillin', 'Morphine')

ORDER BY

allergies,

first\_name,

last\_name;

**or,**

SELECT

first\_name, last\_name, allergies

FROM

patients

WHERE

allergies = 'Penicillin' OR allergies = 'Morphine'

ORDER BY

allergies ASC,

first\_name ASC,

last\_name ASC;

**Questions 8:**

Show patient\_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

**Solution:**

SELECT

patient\_id,

diagnosis

FROM admissions

GROUP BY diagnosis, patient\_id

HAVING COUNT (diagnosis) > 1;

**Questions 9:**

Show the city and the total number of patients in the city.

Order from most to least patients and then by city name ascending.

**Solution:**

SELECT city, COUNT (patient\_id) AS num\_patients

FROM patients

GROUP BY city

ORDER BY COUNT (patient\_id) DESC, city ASC;

**Questions 10:**

Show first name, last name and role of every person that is either patient or doctor.

The roles are either "Patient" or "Doctor".

**Solution:**

SELECT first\_name, last\_name, 'Patient' AS role FROM patients

UNION ALL

SELECT first\_name, last\_name, 'Doctor' AS role FROM doctors;

**Questions 11:**

Show all allergies ordered by popularity. Remove NULL values from query.

**Solution:**

SELECT allergies, COUNT (\*) AS total\_diagnosis

FROM patients

WHERE allergies IS NOT NULL

GROUP BY allergies

ORDER BY total\_diagnosis DESC;

**or,**

SELECT

allergies,

COUNT (\*)

FROM patients

WHERE allergies NOT NULL

GROUP BY allergies

ORDER BY COUNT (\*) DESC;

**or,**

SELECT

allergies,

COUNT (allergies) AS total\_diagnosis

FROM patients

GROUP BY allergies

HAVING

allergies IS NOT NULL

ORDER BY total\_diagnosis DESC;

**Questions 12:**

Show all patient's first\_name, last\_name, and birth\_date who were born in the 1970s decade. Sort the list starting from the earliest birth\_date.

**Solution:**

SELECT first\_name, last\_name,birth\_date

FROM patients

WHERE YEAR (birth\_date) BETWEEN 1970 AND 1979

ORDER BY birth\_date ASC;

**or,**

SELECT

first\_name, last\_name, birth\_date

FROM patients

WHERE

birth\_date >= '1970-01-01' AND birth\_date < '1980-01-01'

ORDER BY birth\_date ASC;

**or,**

SELECT

first\_name,

last\_name,

birth\_date

FROM patients

WHERE YEAR (birth\_date) LIKE '197%'

ORDER BY birth\_date ASC;

**Questions 13:**

We want to display each patient's full name in a single column. Their last\_name in all upper letters must appear first, then first\_name in all lower case letters. Separate the last\_name and first\_name with a comma. Order the list by the first\_name in decending order

EX: SMITH,jane

**Solution:**

SELECT UPPER (last\_name) ||','|| LOWER (first\_name) AS new\_name\_format

FROM patients

ORDER BY first\_name DESC;

**or,**

SELECT

CONCAT (UPPER (last\_name), ',', LOWER (first\_name)) AS new\_name\_format

FROM patients

ORDER BY first\_name DESC;

**Questions 14:**

Show the province\_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000.

**Solution:**

SELECT province\_id, SUM (height) AS sum\_height

FROM patients

GROUP BY province\_id

HAVING SUM (height)>=7000

ORDER BY sum\_height DESC;

**Questions 15:**

Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'.

**Solution:**

SELECT MAX (weight) - MIN (weight) AS weight\_delta

FROM patients

WHERE last\_name = 'Maroni';

**Questions 16:**

Show all of the days of the month (1-31) and how many admission\_dates occurred on that day. Sort by the day with most admissions to least admissions.

**Solution:**

SELECT DAY (admission\_date) AS day\_number,

COUNT (admission\_date) AS number\_of\_admissions

FROM admissions

GROUP BY DAY (admission\_date)

ORDER BY number\_of\_admissions DESC;

**Questions 17:**

Show all columns for patient\_id 542's most recent admission\_date.

**Solution:**

SELECT \*

FROM admissions

WHERE patient\_id = '542'

ORDER BY admission\_date DESC

LIMIT 1;

**or,**

SELECT \*

FROM admissions

WHERE patient\_id = 542

GROUP BY patient\_id

HAVING

admission\_date = MAX (admission\_date);

**or,**

SELECT \*

FROM admissions

WHERE

patient\_id = '542'

AND admission\_date = (

SELECT MAX (admission\_date)

FROM admissions

WHERE patient\_id = '542'

);

**or,**

SELECT \*

FROM admissions

GROUP BY patient\_id

HAVING

patient\_id = 542

AND MAX (admission\_date)

**Questions 18:**

Show patient\_id, attending\_doctor\_id, and diagnosis for admissions that match one of the two criteria:

1. patient\_id is an odd number and attending\_doctor\_id is either 1, 5, or 19.

2. attending\_doctor\_id contains a 2 and the length of patient\_id is 3 characters.

**Solution:**

SELECT

patient\_id,

attending\_doctor\_id,

diagnosis

FROM admissions

WHERE

(

attending\_doctor\_id IN (1, 5, 19)

AND patient\_id % 2 != 0

)

OR

(

LEN (patient\_id)=3

AND

attending\_doctor\_id LIKE '%2%'

);

**Questions 19:**

Show first\_name, last\_name, and total number of admissions attended for each doctor.

Every admission has been attended by a doctor.

**Solution:**

SELECT doctors.first\_name, doctors.last\_name, COUNT (admission\_date)

FROM doctors

JOIN admissions ON admissions.attending\_doctor\_id = doctors.doctor\_id

GROUP BY attending\_doctor\_id;

**or,**

SELECT

first\_name,

last\_name,

COUNT (\*)

FROM

doctors p,

admissions a

where

a.attending\_doctor\_id = p.doctor\_id

GROUP BY p.doctor\_id;

**Questions 20:**

For each doctor, display their id, full name, and the first and last admission date they attended.

**Solution:**

SELECT

doctors.doctor\_id,

doctors.first\_name||' '||doctors.last\_name AS full\_name,

MIN (admissions.admission\_date) AS first\_admission\_date,

MAX (admissions.admission\_date) AS last\_admission\_date

FROM doctors

JOIN admissions on doctors.doctor\_id = admissions.attending\_doctor\_id

GROUP BY admissions.attending\_doctor\_id;

**Questions 21:**

Display the total amount of patients for each province. Order by descending.

**Solution:**

SELECT

province\_names.province\_name,

COUNT (patients.patient\_id) AS patient\_count

FROM province\_names

JOIN patients ON patients.province\_id = province\_names.province\_id

GROUP BY patients.province\_id

ORDER BY patient\_count DESC;

**Questions 22:**

For every admission, display the patient's full name, their admission diagnosis, and their doctor's full name who diagnosed their problem.

**Solution:**

SELECT

p.first\_name || ' ' || p.last\_name as patients\_name,

a.diagnosis,

d.first\_name||' '||d.last\_name as doctor\_name

FROM admissions a

JOIN patients p ON p.patient\_id = a.patient\_id

JOIN doctors d ON d.doctor\_id = a.attending\_doctor\_id;

**Questions 23:**

Display the number of duplicate patients based on their first\_name and last\_name.

**Solution:**

SELECT first\_name, last\_name, COUNT (\*) AS num\_of\_duplicates

FROM patients

GROUP BY first\_name, last\_name

HAVING COUNT (first\_name and last\_name) = 2;

**Questions 24:**

Display patient's full name, height in the units feet rounded to 1 decimal,

weight in the unit pounds rounded to 0 decimals, birth\_date, gender non-abbreviated.

Convert CM to feet by dividing by 30.48.

Convert KG to pounds by multiplying by 2.205.

**Solution:**

SELECT

first\_name || ' ' || last\_name AS patient\_name,

ROUND (height / 30.48, 1) AS 'height "Feet"',

ROUND (weight \* 2.205, 0) AS 'weight "Pound"',

birth\_date,

CASE

WHEN gender = 'M' THEN 'MALE'

ELSE 'FEMALE'

END AS gender

FROM patients;

**Hard Questions**

**Questions 1:**

Show all of the patients grouped into weight groups.

Show the total amount of patients in each weight group.

Order the list by the weight group descending.

For example, if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

**Solution:**

SELECT

COUNT (patient\_id) AS patients\_in\_group,

FLOOR (weight/10)\*10 AS weight\_group

FROM patients

GROUP BY weight\_group

ORDER BY weight\_group DESC;

**Or,**

SELECT

TRUNCATE (weight, -1) AS weight\_group,

COUNT (\*)

FROM patients

GROUP BY weight\_group

ORDER BY weight\_group DESC;

**Or,**

SELECT

COUNT (patient\_id),

weight - weight % 10 AS weight\_group

FROM patients

GROUP BY weight\_group

ORDER BY weight\_group DESC;

**Questions 2:**

Show patient\_id, weight, height, isObese from the patients table.

Display isObese as a boolean 0 or 1.

Obese is defined as weight(kg)/(height(m)2) >= 30.

weight is in units kg.

height is in units cm.

**Solution:**

SELECT

patient\_id,

weight,

height,

CASE

WHEN (weight/power((height\*.01),2))>=30 THEN 1

ELSE 0

END AS isObese

FROM patients;

**Or,**

SELECT

patient\_id,

weight,

height,

weight / POWER (CAST (height AS float) / 100, 2) >= 30 AS obese

FROM patients;

**Questions 3:**

Show patient\_id, first\_name, last\_name, and attending doctor's specialty.

Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'

Check patients, admissions, and doctors tables for required information.

**Solution:**

SELECT

p.patient\_id as patient\_id,

p.first\_name as patient\_first\_name,

p.last\_name as patient\_last\_name,

d.specialty as attending\_doctor\_speciality

FROM admissions a

JOIN patients p ON p.patient\_id = a.patient\_id

JOIN doctors d ON a.attending\_doctor\_id = d.doctor\_id

WHERE a.diagnosis = 'Epilepsy' AND d.first\_name = 'Lisa';

**Or,**

SELECT

pa.patient\_id,

pa.first\_name,

pa.last\_name,

ph1.specialty

FROM patients AS pa

JOIN (

SELECT \*

FROM admissions AS a

JOIN doctors AS ph ON a.attending\_doctor\_id = ph.doctor\_id

) AS ph1 USING (patient\_id)

WHERE

ph1.diagnosis = 'Epilepsy'

AND ph1.first\_name = 'Lisa';

**Or,**

SELECT

a.patient\_id,

a.first\_name,

a.last\_name,

b.specialty

FROM

patients a,

doctors b,

admissions c

WHERE

a.patient\_id = c.patient\_id

AND c.attending\_doctor\_id = b.doctor\_id

AND c.diagnosis = 'Epilepsy'

AND b.first\_name = 'Lisa';

**Or,**

WITH patient\_table AS (

SELECT

patients.patient\_id,

patients.first\_name,

patients.last\_name,

admissions.attending\_doctor\_id

FROM patients

JOIN admissions ON patients.patient\_id = admissions.patient\_id

WHERE

admissions.diagnosis = 'Epilepsy'

)

SELECT

patient\_table.patient\_id,

patient\_table.first\_name,

patient\_table.last\_name,

doctors.specialty

FROM patient\_table

JOIN doctors ON patient\_table.attending\_doctor\_id = doctors.doctor\_id

WHERE doctors.first\_name = 'Lisa';

**Questions 4:**

All patients who have gone through admissions, can see their medical documents on our site. Those patients are given a temporary password after their first admission. Show the patient\_id and temp\_password.

The password must be the following, in order:

1. patient\_id

2. the numerical length of patient's last\_name

3. year of patient's birth\_date

**Solution:**

SELECT

DISTINCT p.patient\_id,

CONCAT (

p.patient\_id,

LEN (p.last\_name),

YEAR (p.birth\_date)

) AS temp\_password

FROM patients p

JOIN admissions a ON a.patient\_id = p.patient\_id;

**Or,**

SELECT

DISTINCT p.patient\_id,

p.patient\_id || FLOOR (LEN (last\_name)) || FLOOR (YEAR (birth\_date)) AS temp\_password

FROM patients p

JOIN admissions a ON p.patient\_id = a.patient\_id

**Or,**

SELECT

pa.patient\_id,

ad.patient\_id || FLOOR (LEN (pa.last\_name)) || FLOOR (YEAR (pa.birth\_date)) AS temp\_password

FROM patients pa

JOIN admissions a ON p.patient\_id = a.patient\_id

GROUP BY pa.patient\_id;

**Questions 5:**

Each admission costs $50 for patients without insurance, and $10 for patients with insurance. All patients with an even patient\_id have insurance.

Give each patient a 'Yes' if they have insurance, and a 'No' if they don't have insurance. Add up the admission\_total cost for each has\_insurance group.

**Solution:**

SELECT

CASE

WHEN patient\_id % 2 = 0 THEN 'Yes'

ELSE 'No'

END AS has\_insurance,

SUM (

CASE

WHEN patient\_id % 2 = 0 THEN 10

ELSE 50

END

) AS cost\_after\_insurance

FROM admissions

GROUP BY has\_insurance;

**Or,**

SELECT 'No' AS has\_insurance, COUNT (\*) \* 50 AS cost

FROM admissions WHERE patient\_id % 2 = 1 GROUP BY has\_insurance

UNION

SELECT 'Yes' AS has\_insurance, COUNT (\*) \* 10 AS cost

FROM admissions WHERE patient\_id % 2 = 0 GROUP BY has\_insurance;

**Or,**

SELECT has\_insurance,

CASE

WHEN has\_insurance = 'Yes' THEN COUNT (has\_insurance) \* 10

ELSE COUNT (has\_insurance) \* 50

END AS cost\_after\_insurance

FROM (

SELECT

CASE

WHEN patient\_id % 2 = 0 THEN 'Yes'

ELSE 'No'

END AS has\_insurance

FROM admissions)

GROUP BY has\_insurance;

**Or,**

SELECT has\_insurance, SUM (admission\_cost) AS admission\_total

FROM

(

SELECT patient\_id,

CASE WHEN patient\_id % 2 = 0 THEN 'Yes' ELSE 'No' END AS has\_insurance,

CASE WHEN patient\_id % 2 = 0 THEN 10 ELSE 50 END AS admission\_cost

FROM admissions)

GROUP BY has\_insurance;

**Questions 6:**

Show the provinces that has more patients identified as 'M' than 'F'. Must only show full province\_name.

**Solution:**

SELECT pr.province\_name

FROM province\_names pr

JOIN patients pt ON pr.province\_id = pt.province\_id

GROUP BY province\_name

HAVING

SUM (CASE WHEN gender = 'M' THEN 1 ELSE 0 END) > SUM (CASE WHEN gender = 'F' THEN 1 ELSE 0 END);

**Or,**

SELECT province\_name

FROM (

SELECT

province\_name,

SUM (gender = 'M') AS n\_male,

SUM (gender = 'F') AS n\_female

FROM patients pa

JOIN province\_names pr ON pa.province\_id = pr.province\_id

GROUP BY province\_name

)

WHERE n\_male > n\_female;

**Or,**

SELECT pr.province\_name

FROM patients AS pa

JOIN province\_names AS pr ON pa.province\_id = pr.province\_id

GROUP BY pr.province\_name

HAVING

SUM (gender = 'M') > SUM (gender = 'F');

**Or,**

SELECT province\_name

FROM patients p

JOIN province\_names r ON p.province\_id = r.province\_id

GROUP BY province\_name

HAVING

SUM (CASE WHEN gender = 'M' THEN 1 ELSE -1 END) > 0;

**Or,**

SELECT pr.province\_name

FROM patients AS pa

JOIN province\_names AS pr ON pa.province\_id = pr.province\_id

GROUP BY pr.province\_name

HAVING

COUNT (CASE WHEN gender = 'M' THEN 1 END) > COUNT (\*) \* 0.5;

**Questions 7:**

We are looking for a specific patient. Pull all columns for the patient who matches the following criteria:

- First\_name contains an 'r' after the first two letters.

- Identifies their gender as 'F'

- Born in February, May, or December

- Their weight would be between 60kg and 80kg

- Their patient\_id is an odd number

- They are from the city 'Kingston'

**Solution:**

SELECT \*

FROM patients

WHERE (first\_name LIKE '\_\_r%')

AND (gender = 'F')

AND (MONTH (birth\_date) = 2 OR MONTH (birth\_date) = 5 OR MONTH (birth\_date) = 12)

AND (weight BETWEEN 60 AND 80)

AND (patient\_id % 2 = 1)

AND (city = 'Kingston');

**Or,**

SELECT \*

FROM patients

WHERE

first\_name LIKE '\_\_r%'

AND gender = 'F'

AND MONTH (birth\_date) IN (2, 5, 12)

AND weight BETWEEN 60 AND 80

AND patient\_id % 2 = 1

AND city = 'Kingston';

**Questions 8:**

Show the percent of patients that have 'M' as their gender. Round the answer to the nearest hundredth number and in percent form.

**Solution:**

SELECT

CONCAT (ROUND ( (

SUM (CASE

WHEN gender = 'M' THEN 1

ELSE NULL

END) / CAST (

COUNT (\*) AS float) \* 100), 2), '%') AS percent\_of\_male\_patients

FROM patients;

**Or,**

SELECT CONCAT (ROUND (

(

SELECT COUNT (\*)

FROM patients

WHERE gender = 'M'

) / CAST (COUNT (\*) AS float),4) \* 100, '%') AS percent\_of\_male\_patients

FROM patients;

**Or,**

SELECT

ROUND (100 \* AVG (gender = 'M'), 2) || '%' AS percent\_of\_male\_patients

FROM patients;

**Or,**

SELECT

CONCAT (ROUND (SUM (gender='M') / CAST (COUNT (\*) AS float), 4) \* 100, '%')

FROM patients;

**Questions 9:**

For each day display the total amount of admissions on that day. Display the amount changed from the previous date.

**Solution:**

SELECT admission\_date,

COUNT (admission\_date) AS admission\_day,

(COUNT (\*) – LAG (COUNT (\*)) OVER (ORDER BY admission\_date)) AS admissions\_change

FROM admissions

GROUP BY admission\_date;

**Or,**

WITH admission\_counts\_table AS (

SELECT admission\_date, COUNT (patient\_id) AS admission\_count

FROM admissions

GROUP BY admission\_date

ORDER BY admission\_date DESC

)

SELECT

admission\_date,

admission\_count,

admission\_count – LAG (admission\_count) OVER (ORDER BY admission\_date) AS admission\_count\_change

FROM admission\_counts\_table;

**Questions 10:**

Sort the province names in ascending order in such a way that the province 'Ontario' is always on top.

**Solution:**

SELECT province\_name

FROM province\_names

ORDER BY (CASE WHEN province\_name='Ontario' THEN 0 ELSE 1 END),

province\_name ASC;

**Or,**

SELECT province\_name

FROM province\_names

ORDER BY

(NOT province\_name = 'Ontario'),

province\_name;

**Or,**

SELECT province\_name

FROM province\_names

ORDER BY

province\_name = 'Ontario' DESC,

province\_name;

**Questions No:**

**Solution:**